WHAT IS CLAIMED IS:

1. A pipelined data processor capable of predicated instruction execution dependent upon the state of an instruction designated predicate register comprising:

a data register file including a plurality of read/write, general purpose data registers;

an instruction decode unit operative during an instruction decode pipeline phase receiving fetched instructions and determining the identity of at least one source operand data register, a destination operand data register and one of a plurality of functional units for execution of each instruction, said instruction decode unit further identifying a predicate register responsive to receipt of a predicated instruction;

a plurality of functional units operative during an execution pipeline phase connected to said instruction decode unit for performing a data processing operation on at least one source operand recalled from at least one corresponding instruction designated source data register and producing a result, said functional unit responsive to a predicate instruction to write said result to an instruction designated destination data register if said corresponding predicate data register has a first state and to nullify said instruction and not write said result if said predicate register has a second state opposite to said first state;

a scoreboard bit corresponding to each data register capable of serving as a predicate register, each scoreboard bit connected to said instruction decode unit to be set to a first digital state upon determining said corresponding data

register is a destination for an instruction and connected to said plurality of functional units to be reset to a second digital state opposite to said first digital state upon functional unit write of a result to said corresponding data register; and

each functional unit is further operative responsive to a predicate instruction during a decode pipeline phase to nullify said predicate instruction of a following execution phase if said predicate register has said second state and said corresponding scoreboard bit has said second state.

2. The pipelined data processor of claim 1, wherein:

said functional unit is further operative to reset said scoreboard bit to said second digital state upon nullification of said instruction designating a corresponding data register as a destination operand data register.

3. The pipelined data processor of claim 1, wherein:

each functional unit is further operative to operate at a reduced power state for any execution pipeline phase nullified during said preceding instruction decode phase.

4. A method of operating a pipelined data processor capable of predicated instruction execution dependent upon the state of an instruction designated predicate register comprising the steps of:

setting a scoreboard bit to a first digital state upon determining a corresponding data register is a destination for an instruction;

resetting a scoreboard bit to a second digital state opposite to said first digital state upon a write of a result to said corresponding data register;

performing a data processing operation on at least one source operand recalled from at least one corresponding instruction designated source data register and producing a result in response to a predicate instruction designating a corresponding predicate data register and writing said result to an instruction designated destination data register if said corresponding predicate data register has a first state;

nullifying a predicate instruction and not writing said result to the instruction designated destination data register if said corresponding predicate register has a second state opposite to said first state;

nullifying a predicate instruction for a following execution phase if said corresponding predicate register has said second state and said corresponding scoreboard bit has said second state during a prior decode phase.

5. The method of claim 4, further comprising the step of:

resetting a scoreboard bit to a second digital state upon nullification of said instruction designating said corresponding data register as a destination operand data register.

6. The method of claim 4, further comprising the step of:

operating the functional unit at a reduced power state for any execution pipeline phase nullified during said preceding instruction decode phase.

7. The method of claim 4 further comprising the step of:

scheduling a last write to a data register a predetermined number of pipeline phases before an execution phase of a predicate instruction designating said data register as a predicate register.